AIAA 99-3303

ADVANCES IN AUTOMATIC OVERSET GRID GENERATION AROUND SURFACE DISCONTINUITIES

William M. Chan
MCAT, Inc., Army/NASA Rotorcraft Division,
NASA Ames Research Center,
Moffett Field, California 94035

Reynaldo J. Gomez III
NASA Johnson Space Center,
Houston,Texas 77058

AIAA 14th Computational Fluid Dynamics Conference
June 28 - July 1, 1999 / Norfolk, VA

AIAA 99-3303

ADVANCES IN AUTOMATIC OVERSET GRID GENERATION AROUND SURFACE DISCONTINUITIES

William M. Chan *
MCAT, Inc., Army/NASA Rotorcraft Division,
NASA Ames Research Center,
Moffett Field, California 94035

Reynaldo J. Gomez III †
NASA Johnson Space Center,
Houston, Texas 77058

Abstract

A scheme is described for the automatic domain decomposition and creation of overset grids around surface discontinuities. Curves along surface discontinuities are called seam curves. The seam curves are first automatically extracted from a multiple panel network description of the surface. Points where three or more seam curves meet are automatically identified and are called seam corners. Surface grids that cover the local region around the seam corners are then automatically generated. The seam curves are automatically trimmed away from the seam corners, and hyperbolic surface grids are grown from the trimmed seam curves. Examples of complex grid systems created with the above scheme are given for the X-38 Crew Return Vehicle, a generic finned-store configuration, and the Hyper-X research vehicle. Steady and unsteady flow solutions are computed for the first two cases, respectively, and the computed flow results are compared with those obtained using more conventional grid systems.

1. Introduction

In the overset grid approach[1] to computing flows around complex configurations, surface grid generation remains a time consuming task. The process involves decomposition of the surface domain and creation of overlapping surface grids; both always require some user expertise and significant effort. This paper describes a scheme that reduces the manual labor needed to perform the above tasks. Such improvements of the overset computational procedure can significantly enhance the design and analysis process of complex aerospace vehicles. The current work is built upon two earlier efforts involving overset surface grid generation. A brief review will first be given.

The first effort is the concept of hyperbolic surface grid generation introduced by Steger.[2,3] A nearly orthogonal surface grid is created by marching from an initial curve onto a reference surface. The side and outer boundaries do not have to be specified, and can remain unconstrained. Such methods are particularly suited for the overset approach where neighboring grids can overlap arbitrarily. Moreover, grid generation effort is reduced since only one initial curve needs to be specified instead of the four initial curves required for algebraic and elliptic methods. Extension of Steger's scheme to march on complex reference surfaces consisting of multiple panel networks was performed by Chan and Buning.[4] Multiple surface grids are marched onto the reference surface from a set of initial curves. Limited control of the behavior at the side boundaries is available through user-specified boundary conditions. Such a collection of hyperbolic surface grids may or may not cover the reference surface completely.

The second effort was performed by Chan and Meakin[5] to automatically fill the gaps on the reference surface not covered by the set of hyperbolically grown surface grids. This work also introduced the terminology of seam curves and seam grids. Seam curves are curves on the geometry surface where a grid line should be placed. These include curves along sharp edges, intersection curves between components, high curvature contours such as leading edges of wings, and open boundaries such as the symmetry plane curve of a half body. Surface grids grown from the seam curves by a marching scheme are called seam grids. A procedure was developed to automatically fill the gaps between the seam grids on the smooth regions of the surface using algebraic methods.

The current work contributes to the automation

[*] Senior Research Scientist, MCAT, Inc., Senior Member AIAA

[†] Aerospace Engineer, NASA Johnson Space Center, Member AIAA

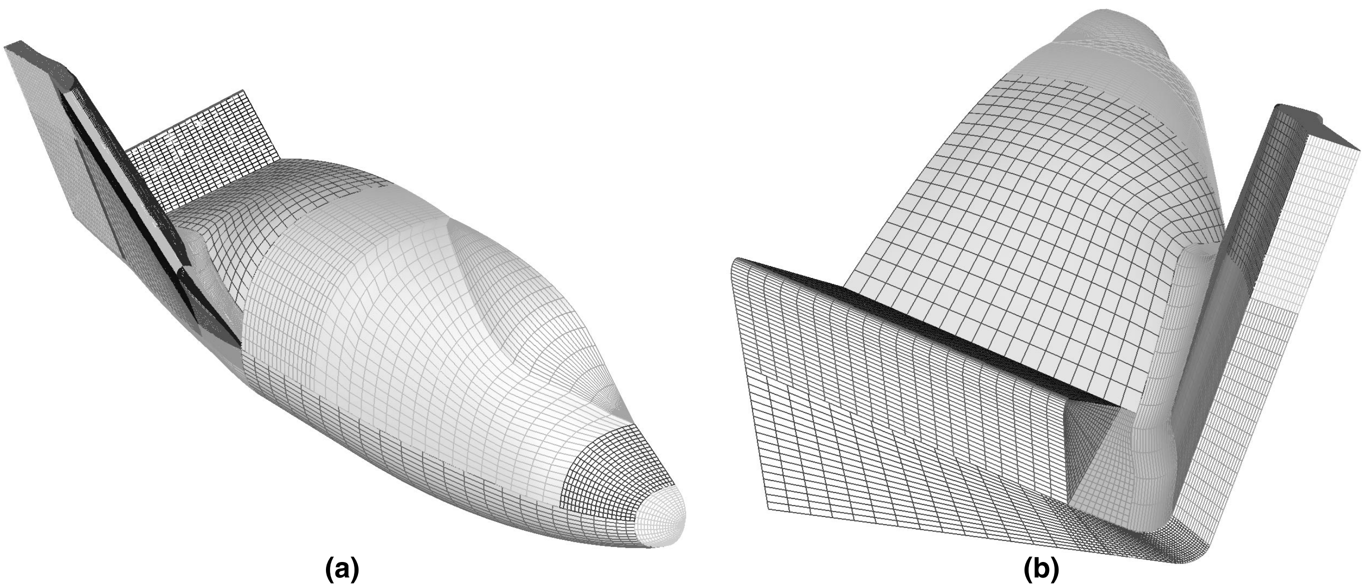

Fig. 1 X-38 panel network geometry definition. (a) Front view. (b) Back view

of the procedure required in generating seam grids. Previously, seam curves needed to be manually identified and extracted from the surface definition using a CAD package (e.g., ICEMCFD,[6] PRO-ENGINEER[7]), or a grid generation package (e.g., GRIDGEN,[8] OVERGRID[9]). The user had to then decompose the region around the seam curves into four-sided domains for the creation of seam grids. This step typically relied heavily on the experience of the user on finding proper strategies to perform decomposition on complex surface domains. The difficulty lay in the fact that there is no unique decomposition of the surface and that there are no formal rules for the construction of a 'good' decomposition. Grid points then had to be distributed on the seam curves to properly resolve the local geometry. Creation of the seam surface grids by hyperbolic marching also required the specification of marching distances and grid spacings. A scheme to automate most of the above steps, including some simple rules for domain decomposition around surface discontinuities is described in Section 2. A solution pre-processing tool for automatic creation of input files for the volume grid generator, domain connectivity program and flow solver is presented in Section 3. Some results on flow solutions obtained using the current new gridding strategy are shown in Section 4. Concluding remarks are given in Section 5.

2. Surface Grid Generation Scheme

The current scheme for overset seam grid generation consists of 5 automated steps which are described in Sections 2.1 to 2.5 below. The X-38 Crew Return Vehicle is used as an example to illustrate the various steps. Section 2.6 presents a more complex example on the Hyper-X vehicle. A software tool called SEAMCR was developed to perform these steps.

2.1. Seam Curve Identification and Extraction

The starting point of the process is the surface geometry of the configuration. Currently, only a multiple panel network description of the surface is accepted (see Figure 1). Each panel network consists of a rectangular array of points. A degenerate edge on a boundary of a panel network is allowed where all points collapse to a singular point. Small gaps up to a given tolerance are permitted between panel networks, but gaps are not allowed near points where multiple seam curves meet. Also, the surface description is assumed to be trimmed, i.e., only the outer mold line or wetted surface of the geometry is present.

Given the above constraints on the surface description, the first step of the process is to identify the seam curves automatically. A seam curve is made up of a sequence of segments where each segment is an edge of a quadrilateral (quad) on the panel network description of the surface. An interior or boundary edge of a panel network may be selected. The following three situations may arise.

(1) An interior edge is selected if the angle between the surface normals of the two quads it separates is larger than a given angle threshold (Figure 2a).

(2) A boundary edge is selected if it is adjacent to no other panel network, i.e., it is on an open boundary such as the symmetry plane for a half vehicle (Figure 2b).

(3) A search is performed to determine if a boundary edge is adjacent to a quad from a neighboring panel network. The boundary edge is selected if it passes the above angle test for the two quads that it is adjacent to in panel networks A and B as shown in the example in Figure 2b.

The quad edges selected by the above criteria are automatically connected to form a set of seam curves. If

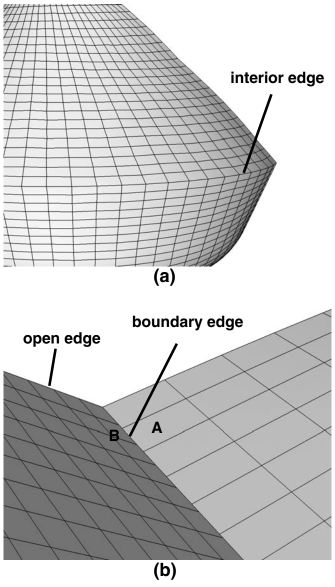

Fig. 2 Seam curve picking criteria.

the surface definition is too coarse, the panel networks may not sufficiently resolve the surface curvature and the above scheme may identify some extraneous seam curves in the coarse regions. Adjustment of the angle threshold in cases (1) and (3) above may not be enough to avoid such situations. In these cases, the user can use an interactive graphical tool such as OVERGRID to remove the extraneous seam curves before proceeding to the next step. The SEAMCR code provides a mechanism to stop the process after the seam curve extraction step. After adjustments of the seam curves by the user, the results can be fed to SEAMCR in restart mode to complete the rest of the steps.

Adjusted seam curves for the X-38 are shown in Figure 3. On inspection, there is no obvious best choice on an approach for generating hyperbolic surface grids from these seam curves. The difficulty lies in automatically determining the end-point treatment where multiple seam curves meet. However, automation may be more attainable if the seam curves are trimmed away from the intersection points. Then grid lines emanating from the end points of each seam curve can be allowed to float freely under a hyperbolic marching scheme. The next step then, is to identify the seam curve intersection points (seam corners) as described in the next section.

2.2. Seam Corner Identification

If one end of a seam curve meets an interior point from another seam curve, the latter seam curve is automatically split into 2 seam curves at the intersection point. A seam corner is defined to be a point where 3 or more end points of seam curves coincide; and such points are automatically identified by SEAMCR. The degree of the corner is given by the number of seam curves meeting at the corner. A corner on an open boundary of the configuration is called an open corner, otherwise, it is called a closed corner.

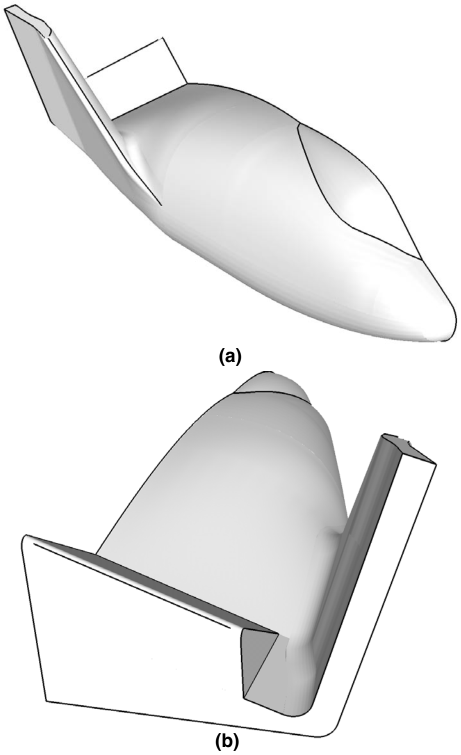

Fig. 3 Seam curves for the X-38. (a) Front view. (b) Back view.

2.3. Seam Corner Surface Grid Generation

In order to build a grid that has grid lines coinciding to all seam curves connected to a seam corner, a special surface grid topology is introduced with a singular axis point at the corner and radial curves emanating from the corner. Each seam curve connected to the seam corner is followed by one of the radial curves of the surface grid. The topology of such a grid is analogous to a polar grid in two dimensions. An automatic procedure for generating such a spider-web-like surface grid is described in this section. The spider web grid generation scheme described below is valid for corners of any degree. For a certain class of seam corners of degree 3 or 4, it is possible to design a surface grid topology that does not involve a singular point. While special tests can be implemented to detect such situations, this paper will focus on an automatic procedure for the spider web topology that works for any type of seam corner.

2.3.1. *Seam Curve Ordering*

The first step is to determine the order of the seam curves going around the corner, and whether the corner is open or closed. An open corner is situated on an open boundary of the surface domain while a closed

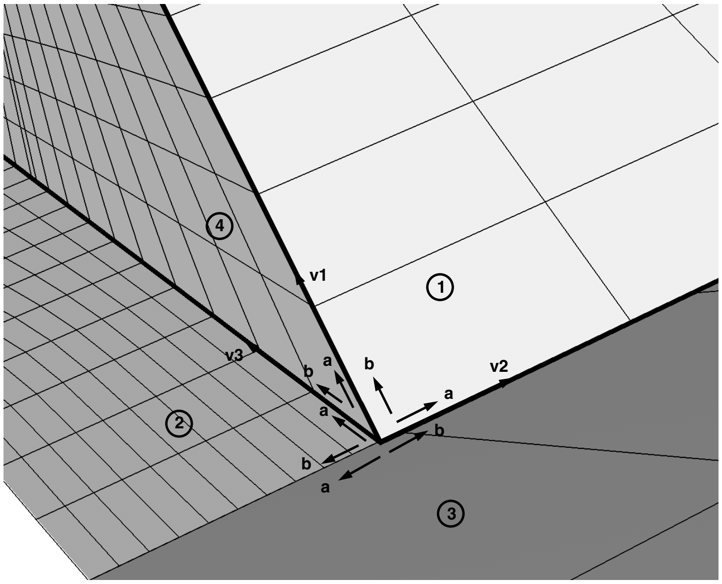

Fig. 4 Vectors used for seam curve ordering scheme around a corner of degree 3. A concave turn exists between patches 1 and 3, and between 2 and 4. A convex turn exists between patches 1 and 4. Patches 2 and 3 are approximately coplanar. Seam curves lie along v1, v2 and v3.

corner is situated in the interior of the surface domain. Figure 4 shows a closed corner of degree 3 surrounded by 4 surface panel networks. The cells from the surrounding panel networks that touch the corner are identified. Two cases may arise. In case 1, the corner coincides with a vertex of a cell as in patches 1, 2 and 4 in Figure 4. In case 2, the corner lies on an edge of a cell as in patch 3 in Figure 4. The next step is to identify unit vectors **a** and **b** for each of these cells. These vectors are parallel to an edge of a cell and point away from the corner. In case 1 above, vectors **a** and **b** are chosen such that $\mathbf{a} \times \mathbf{b}$ is in the direction of the cell surface normal **n**. In case 2 above, **a** and **b** point in opposite directions and **a** is chosen such that $\mathbf{n} \times \mathbf{a}$ points into the cell.

For a closed corner, vector **a** from any cell is parallel to a vector **b** from another cell. A corner is labelled open if there is a vector **a** that cannot be paired with a vector **b** from another cell. The pairing of vectors **a** and **b** also determines the ordering of the cells around the corner, e.g., starting with vector **a** on patch 4, vector **b** on patch 4 is matched with vector **a** from patch 2, vector **b** on patch 2 is matched with vector **a** from patch 3, etc.

Once the cells around the corner are ordered, ordering of the seam curves can now proceed easily. For an open corner, start with the cell that contains the open vector **a** and find the seam curve that matches this vector. For a closed corner, start with any cell whose vector **a** matches a seam curve. Follow the order of the cells until a cell is reached whose vector **a** matches another seam curve. The angle between any two seam curves on the surface is computed by summing the angles between pairs of vectors **a** and **b** traversed in the ordering process.

2.3.2. *Circumferential Point Distribution*

The seam curves divide the circumferential direction into sectors where each sector is bounded by two seam curves. The number of points used in each sector N_s is based on the angle θ_s between the bounding seam curves at the corner. A general rule of one point per 10 degrees is used but the number of points in a sector is not allowed to fall below 5. The distribution of points in the circumferential direction is further readjusted by a scheme discussed in Section 2.3.4.

2.3.3. *Determination of Radial Distance*

A uniform grid spacing is used in the radial direction. This is given by a global parameter Δs_g that governs the desired general grid resolution on the geometry surface. The parameter can be user supplied or given by a default based on some fraction of the bounding box diagonal of the configuration. The actual radial grid spacing used in each spider web grid is adjusted if necessary so that there are at least 5 points in the radial direction.

The radial extent of the surface grid is automatically determined from the smallest length computed from the following 3 constraints.

(1) Fraction of distance from the corner to the closest segment of any seam curve not connected to the corner.

(2) Fraction of the total length of a seam curve connected to the corner.

(3) The smallest $\Delta s_g / \Delta\theta_s$ over all sectors where $\Delta\theta_s = \theta_s/(N_s - 1)$ and all angles are measured in radians.

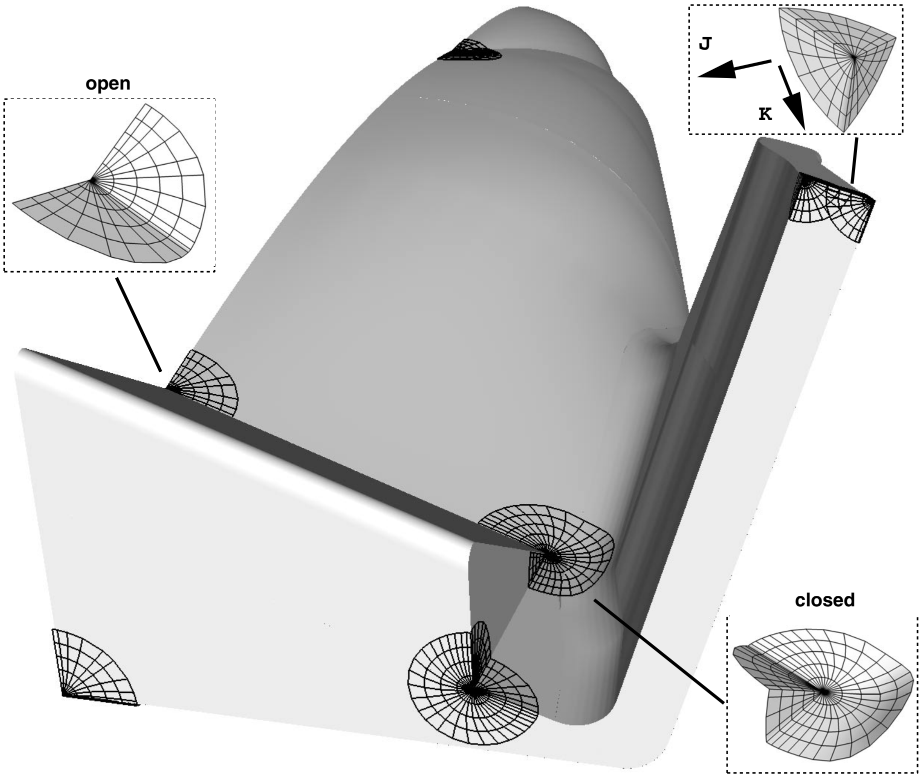

Fig. 5 Spider web grids for the X-38.

Condition (1) ensures that the surface grid does not get too close to neighboring seam curves. Condition (2) ensures that the surface grid does not get too close to the other end of a seam curve connected to the corner. A typical value for the fraction used for both conditions (1) and (2) is about 0.7. Condition (3) is an estimate on the maximum circumferential grid spacing based on the assumption that the seam curves connected to the corner are straight lines. The maximum circumferential grid spacing increases with radial distance from the corner. By limiting this grid spacing to be less than the global spacing parameter Δs_g, the maximum radial extent of the grid is also restricted.

2.3.4. *Generation of Spider Web Grids*

The algorithm used for creating the spider web grids is outlined here. Let the J index denote the radial direction with J=1 at the corner, and the K index denote the circumferential direction.

(1) Points along the seam curves connected to the corner are redistributed based on the uniform spacing and up to the radial distance discussed in Section 2.3.3. The redistributed points are loaded into appropriate K=constant grid lines of the surface grid. Note that the seam curves and hence these grid lines are not necessarily straight lines.

(2) The remaining K=constant grid lines are filled by sector. For each sector, points along a J=constant grid line are obtained by blending the radial vector from the corner and the vectors along the bounding seam curves of the sector. The blended grid points are projected back onto the bilinear surface defined by the multiple panel network of the reference surface. Since the seam curves are not always straight lines, the actual maximum circumferential grid spacing may exceed the estimates used in Sections 2.3.2 and 2.3.3. Such situations are automatically detected, and extra grid points are automatically added and smoothed in the circumferential direction so that Δs_g is not exceeded.

Spider web grids created on the X-38 with the above automatic scheme are shown in Figure 5.

2.4. Trimming and Point Redistribution

The next step requires trimming the seam curves from the seam corners. End points of seam curves connected to a corner are automatically removed such that sufficient overlap is maintained between the spider web grid and the seam curves. Next, grid points are automatically redistributed on the trimmed seam curves based on turning angles such that the global grid spacing Δs_g is never exceeded. The turning angle ϕ_j (radians) at point j on a curve is defined to be

$$\phi_j = \pi - \cos^{-1}(\vec{r}_{j+} \cdot \vec{r}_{j-}) \qquad (1)$$

where $\vec{r}_j = (x, y, z)^T_j$, $\vec{r}_{j+}$ is a unit vector in the direction of $(\vec{r}_{j+1} - \vec{r}_j)$ and $\vec{r}_{j-}$ is a unit vector in the direction of $(\vec{r}_{j-1} - \vec{r}_j)$. For example, ϕ would be zero for segments along a straight line.

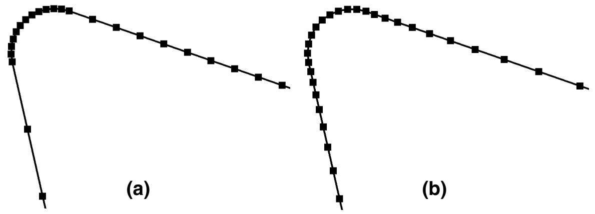

Fig. 6 Automatic point redistribution on a curve based on turning angle. (a) Original point distribution. (b) New point distribution.

Sharp and rounded turns on a seam curve are automatically detected and the grid spacing is reduced to $\alpha\Delta s_g$ around the turn where $0 < \alpha < 1$. Detection of a sharp turn is achieved by simply checking if the turning angle exceeds a specified threshold at a point. A robust method to detect a rounded turn is more difficult to design, however. The current scheme analyzes the turning angles over a range of points to locate the start and end of the turn. Note that the turning angle at each point inside the turn can be different from each other, and its value may even be zero at one of more points in the turn. For a sharp turn, the total turning angle ϕ_t is used to determine α, which is set to $1 - \tanh(\phi_t/90.0)$ with ϕ_t measured in degrees. With this scheme, the grid spacing decreases with increase of turning angle. For a rounded turn, a uniform spacing of $\alpha\Delta s_g$ is employed inside the turn where α is computed such that the turning angle at any point in the redistributed curve does not exceed a given ϕ_{max}. In SEAMCR, a ϕ_{max} of 15 degrees is currently used. A special stretching function was developed to geometrically blend the $\alpha\Delta s_g$ spacing at the turn to a uniform spacing of Δs_g away from the turn. Automatic grid point redistribution near a rounded turn on a seam curve is illustrated in Figure 6.

2.5. Generation of Hyperbolic Surface Grids

The final step is to create the seam surface grids by marching from the trimmed seam curves. For each seam curve, the following inputs are required by the hyperbolic surface grid generator SURGRD.[4] Complete automation is achieved if all 5 inputs are automatically computed.

(1) Marching direction.
(2) Boundary conditions at the end points of the initial curve.
(3) Stretching ratio in marching direction.
(4) Initial and end spacing in marching direction.
(5) Marching distance.

For each initial curve, marching can be performed in only one direction if the curve lies on an open boundary of the geometry; otherwise, marching must be performed in both directions. Item (1) above is easily automated by determining whether the initial curve lies on an open boundary. Since the seam curves have been trimmed from the corners, the grid lines emanating from the end points are free to float along the surface. Hence, item (2) is automated by simply selecting the free floating boundary condition at the end points.

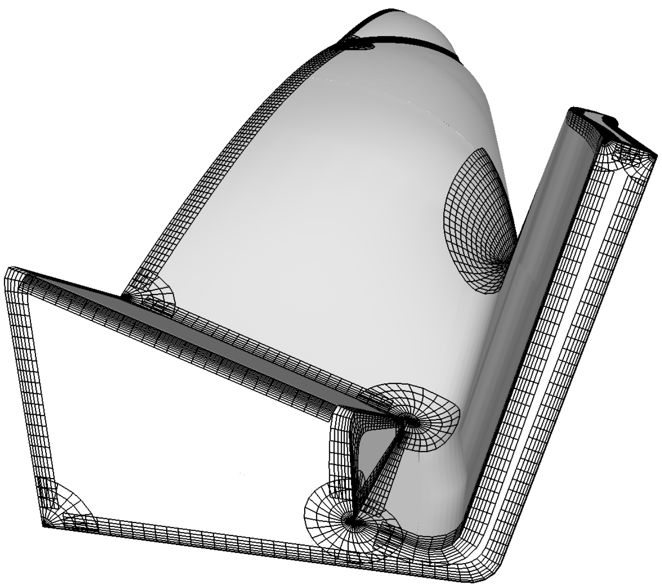

Fig. 7 Seam grids and spider web grids for the X-38.

Items 3, 4 and 5 define the spacing distribution in the marching direction. A new feature in SURGRD specifies the spacing distribution via the input of a stretching ratio, initial and end spacing, and a marching distance. Starting from the initial curve, grid points in the marching direction are created using a geometric stretching with the given stretching ratio and initial spacing. If the specified end spacing is reached before the marching distance is completed, the rest of the domain is padded with a uniform grid made up with the prescribed end spacing. The stretching ratio is automatically adjusted if necessary so that the grid points fit within the marching distance. If the specified end spacing is not reached before the marching distance is completed, the scheme falls back to a hyperbolic tangent stretching with the prescribed initial and end spacing. The advantage of this scheme is as follows.

(1) The same stretching ratio, initial and end spacing can be used for all grids. There is no need to predetermine the number of points needed for the marching distance specified.
(2) By specifying the end spacing to be the global grid spacing parameter Δs_g, the stretching function will never produce a grid that exceeds this spacing.

For items 3 and 4 of the SURGRD input, a stretching ratio of 1.2, an initial spacing of $0.4 \times \Delta s_g$ and an end spacing of Δs_g were used for all grids. Automatic determination of the marching distance is a more difficult problem. In general, each point on the initial curve may be prescribed a different marching distance. For now, the marching distances at the two ends of the initial curve are automatically determined and linear interpolation is used to obtain the marching distances for the points in between. The marching distance at an end point is chosen to be the same as the radial extent of any spider web grid connected to that end of the initial curve. If no spider web grid is connected to an end point, the marching distance is chosen to be half the distance to the closest edge of any neighboring seam curve. A SURGRD input file with the above automated parameter selection is written out by SEAMCR. The result of applying this input file to the X-38 test case is shown in Figure 7.

After following the surface grid generation scheme described in Sections 2.1 to 2.5, one finds that the domain around the surface discontinuities has been decomposed and surface grids have been created around these discontinuities. The resulting set of seam grids can now be handed to the SBLOCK code[5] to fill the gaps in between the seam grids.

2.6. Hyper-X Test Case

The surface grid generation scheme described in this section is applied to a more complex geometry for testing. Figure 8 shows the front, back, and various close-up views of spider web grids and trimmed seam curves for the Hyper-X vehicle. The configuration consists of 51 panel networks and 38 seam curves. A total of 23 spider web grids and 45 trimmed seam curves were created. Several of the spider web grids have to cover complex surface saddle points where sharp convex and concave corners meet. A close up view of the volume grid at one of the saddle points is given in Figure 9.

3. Flow Solution Pre-processing

The procedure described so far has reduced the time and effort needed from the user for generating overset surface grids. Such automated techniques tend to produce a large number of surface grids, increasing the amount of work required to create input files for all of the subsequent steps in the flow computation process. These steps include the manual creation of input files for the volume grid generator, domain connectivity code, flow solver and visualization software and subsequent iterative modifications to these files.

The OVERCHK utility code has been developed to automate the creation of these input files. The new domain decomposition technique generates a collection of surface grids with fairly simple topologies, making boundary condition detection a straightforward procedure. Additionally, these topologically simple surface grids are ideal for hyperbolic volume grid generation and make robust automated volume grid generation possible. OVERCHK automatically detects a number of different boundary conditions from an input set of surface grids or volume grids and uses these boundary conditions to create input files for the HYPGEN[10] hyperbolic volume grid generator, the PEGASUS[11] domain connectivity code, and the OVERFLOW[12] flow solver. Additionally, these boundary conditions can be written out as a PLOT3D[13] command file to visually display and verify the boundary conditions for all of the input grids simultaneously.

4. Flow Solution Results

The new gridding strategy described in Section 2 tends to produce more grids than a typical manual decomposition scheme. Moreover, some of the automatically created grids contain a singular axis point. These singular points are used to resolve sharp geometric discontinuities and often have to resolve important flow field discontinuities. It is important to verify that such a grid system does not degrade the accuracy, stability, and convergence characteristics of the flow solution. Tests performed on two configurations are described below.

4.1. X-38 Crew Return Vehicle

The first example used to test the proposed domain decomposition scheme was the X-38 Crew Return Vehicle, design revision 3.1. The SBLOCK code[5] was first employed to fill the gaps between the seam grids shown in Figure 7. Unfortunately, some of the grids produced were undesirably small. A different approach to fill the gaps was employed by manually adjusting the hyperbolic marching distances from the various seam curves using the OVERGRID[9] graphical interface. Although it is not automated yet, the entire process required only about 3 hours of user time.

Solutions on the present grid system were compared to solutions obtained on the original grid system created at NASA Johnson Space Center. Surface grids for the present and original grid systems are shown in Figure 10. Freestream conditions for this case were: Mach number = 0.8, Alpha = 10.0 degrees, Reynolds number = 112,500/inch. The goal of this example was to perform a detailed comparison of the results with the new gridding scheme versus a manually created decomposition. Starting with a file containing the surface grids, input files were created for the HYPGEN hyperbolic volume grid generator, the domain connectivity code PEGASUS and the OVERFLOW flow solver using the OVERCHK code. Approximately 30 minutes of user time were required to create a volume grid system with no negative Jacobians. The

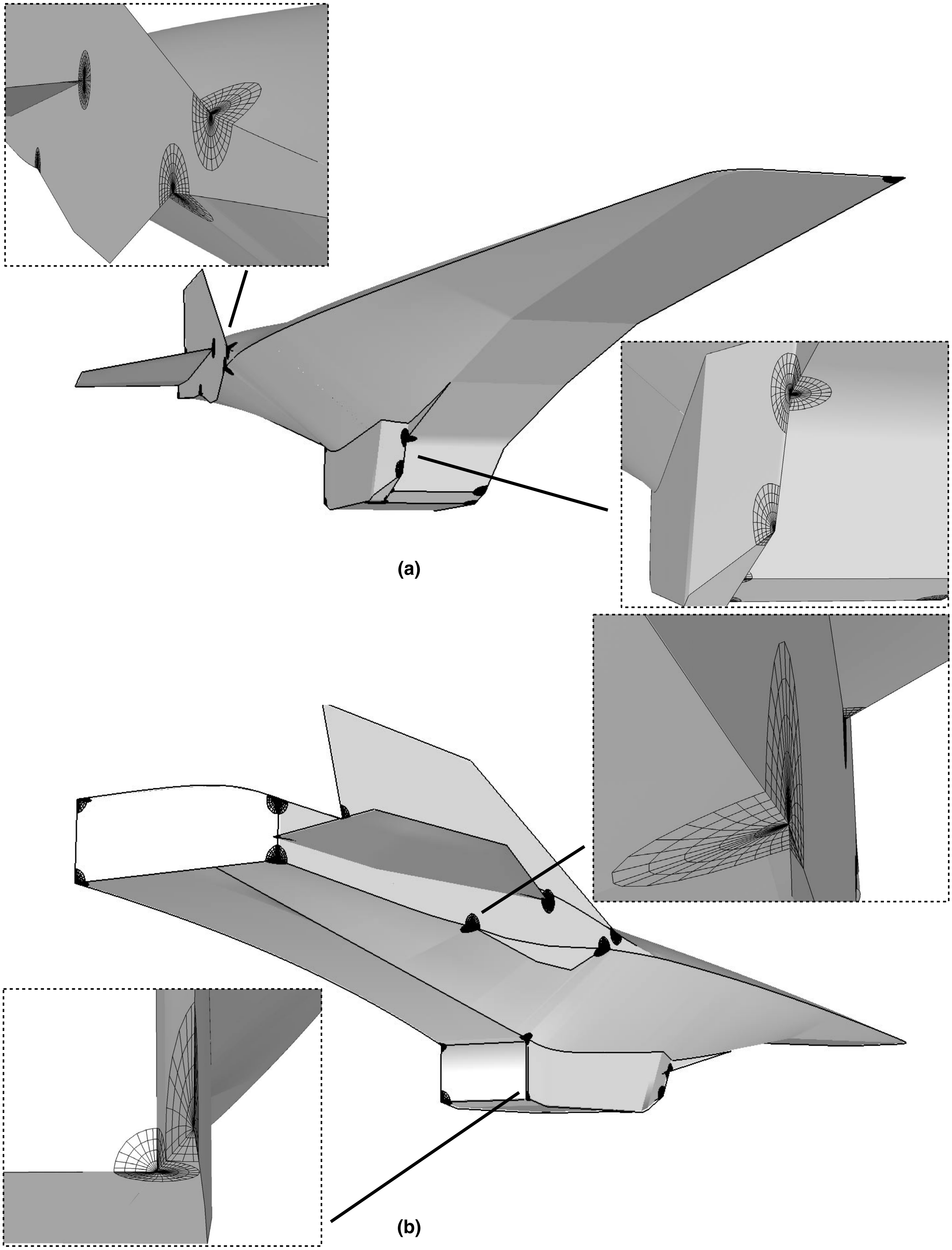

Fig. 8 **Spider web grids and trimmed seam curves for the Hyper-X. (a) Front view. (b) Back view.**

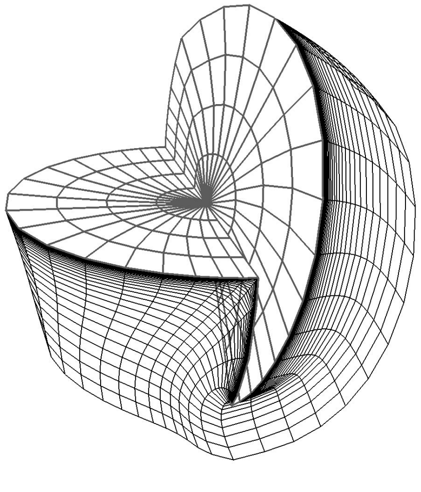

Fig. 9 Slices of a spider web volume grid at a saddle point.

final run to create all volume grids took less than 3 minutes on a Silicon Graphics 195MHz R10000 processor. The Cartesian background grid and far field grid were created using the BOXGR module in OVERGRID. Domain connectivity was performed using the PEGASUS code (version 5.0b3) and required less than 2000 CPU seconds on a single processor of a Silicon Graphics Origin 2000.

The original grid system consists of 14 viscous zones, one Cartesian background grid and a far field grid, with a total of 1.78 million grid points. The present grid system consists of 30 viscous zones, and the same Cartesian background grid and far field grid used in the original grid system, with a total of 1.91 million grid points. Input files for the two grid systems used the same damping values and boundary conditions. Three levels of full multigrid were used to start both solutions. The original system used 100 iterations on each grid level while the present system required 200 iterations on the two coarser levels to successfully run past the initial startup transient. This is due to the smaller grid spacing in the present grids relative the original grid system and the fact that they have explicit boundary communications in high flow gradient regions. The CPU time required to run these additional iterations was less than three percent of the total time required for a solution. Subsequent to the initial startup, the same multigrid steps were used on both grid systems. The pitching moments for both solutions were converged to three decimal places within 450 iterations on the finest grid level. No convergence or stability problems were found in the spider web grids.

Each steady-state solution required approximately 34 hours of Cray J90 CPU time and 40 million words of memory. Another solution for this new grid system was performed at the Numerical Aerodynamic Simulation (NAS) Facility at NASA Ames Research Center with a version of OVERFLOW optimized for the Silicon Graphics Origin 2000. Using 32 processors, this solution was converged in less than 1.5 wall clock hours. Elapsed time for this entire process, from PRO-ENGINEER IGES file to converged solution was about one and a half days for the present system, and three and a half days for the original system. Table 1 compares the time (in days) needed for the various parts of the simulation. Although there are differences in the machine type for the solution process, it shows that the current technology cuts the entire process time by about half.

	Original	Present
Panel network creation	0.5	0.5
Surface & volume grids	2.0	0.5
Domain connectivity	0.5	0.1
Flow solution	0.3(J90)	0.05 (ORIGIN)

Table 1. Comparison of time taken for various parts of the simulation process between original and present systems (time measured in days).

Figure 11 compares the centerline surface pressure coefficient for the two grid systems. In general, the comparison is quite good. The spider web grids in expansion regions appear to cause some minor local oscillations in pressure. Examples of these oscillations are at 125 inches, behind the canopy, and at 275 inches in the expansion around the lower portion of the X-38 base. The slight mismatch in the peak expansion location at 60-80 inches is primarily due to the lack of a seam curve in the original grid system compared to the more accurate geometric representation in the new system. The pressure oscillation on the lower surface near 260 inches appears to be the result of a problem in the original grid system in an overlap region. The spider web grids do not appear to have any adverse effect on the surface pressure distribution.

Integrated forces and moments for the two grid systems are compared in Table 2. Both solutions underpredict C_L and overpredict C_D. The present results tend to underpredict expansion and compression peaks producing a lower C_L value. This underprediction appears to be due to the present grid system's lower resolution in these regions. Both results are within the error bounds of the wind tunnel data.

	Points	C_L	C_D	C_m
Original	1.78×10^6	0.119	0.162	0.055
Present	1.91×10^6	0.117	0.155	0.053
Wind tunnel	N/A	0.127	0.126	0.053

Table 2. Comparison of force and moment coefficients between original and present grid systems, and wind tunnel data.

4.2. Finned-store Configuration

The second example tested was a generic finned-store which was part of a generic wing/pylon/finned-store unsteady flow test case.[14,15] For this work, the

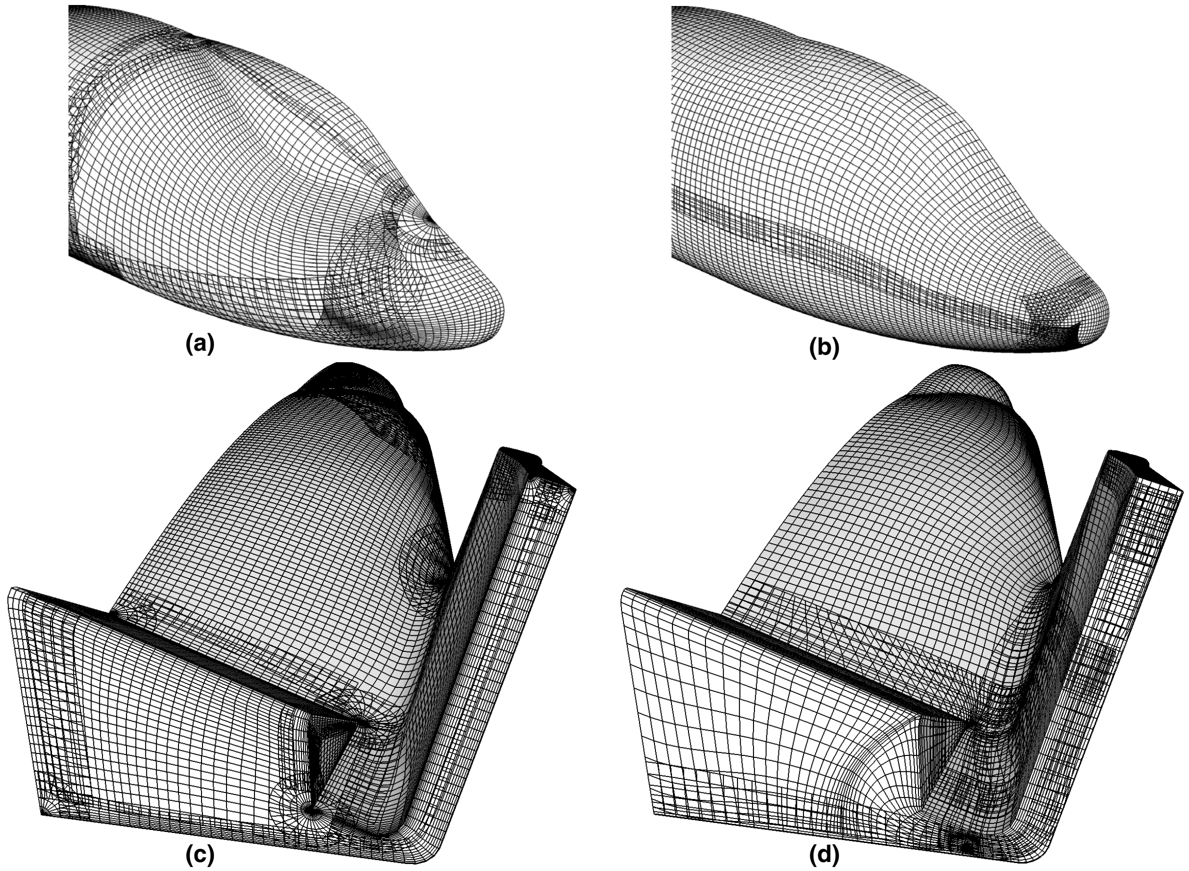

Fig. 10 Present and original grid systems of the X-38. (a) Present system front view. (b) Original system front view. (c) Present system back view. (a) Original system back view.

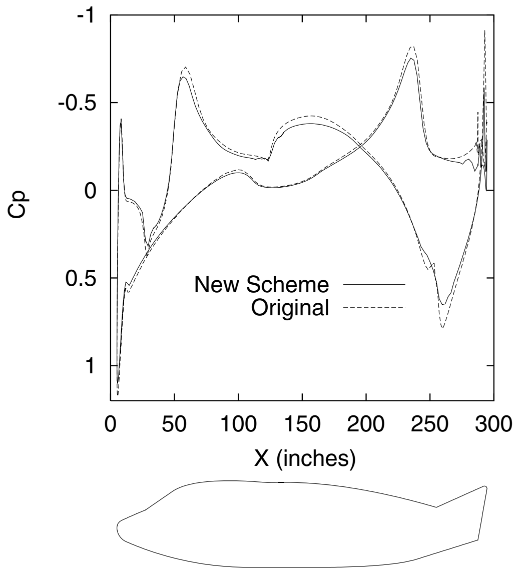

Fig. 11 Surface pressure coefficient on the y = 0 centerline for the X-38.

wing/pylon were not included in the calculation, and time accurate computations were performed on the stationary finned-store. The present and original grid systems are shown in Figure 12. In the original system, a grid with two viscous directions was used to model the fin/store junction. In the present grid system, collar grids[16] plus spider web grids with just one viscous direction were used instead. In this case, the SBLOCK code created satisfactory grids in the gaps between the seam grids.

A steady state solution at Mach 0.6 and zero degree angle of attack was created and used to start the time accurate simulation. The unsteady solution was performed using the OVERFLOW flow solver with 3 Newton subiterations per time step. No problems were found that were related to the singular axis point topology in the spider web grids. However, some problems were encountered with the original grid system's fin grids where viscous spacing occurred in two directions. The flow solver overpredicted the pressure distribution on the leading edge of the fins compared to the present grid system. After modifying the fin grids in the original system to use a collar grid topology with a single viscous direction, better agreement was obtained between the original and present solutions. The same time step was used for the modified original and present solutions but its size was restricted by the less robust modified original system.

The solutions for the modified original system and the present system are compared in Figures 13 and 14. For the unsteady run, it was found that the surface pressure has very little variation with time. Figure 13 compares the centerline (y = 0) surface pressure co-

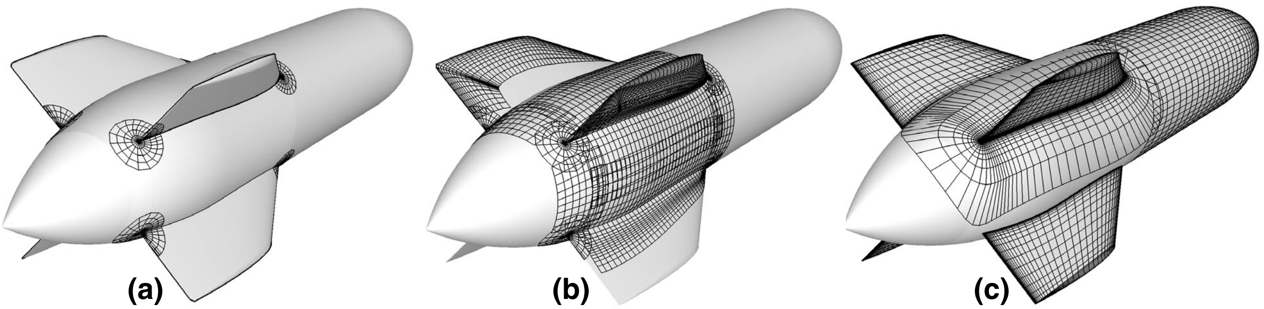

Fig. 12 (a) Spider web grids and trimmed seam curves for finned-store. (b) Sample of surface grids for present finned-store grid system. (c) Sample of surface grids for original finned-store grid system.

efficient for the two grid systems at different time iterations. Despite the difference in the time histories, the plots compare well indicating there may be only small unsteadiness. One source of such unsteadiness is found near the store nose where a stagnation point coincides with the application of axis boundary conditions. Figure 14 compares the surface pressure coefficient along a 45 degree cross-section through the store and fins for the two grid systems. The oscillation near the nose is not as prominent in this cross section cut and the overall agreement is fairly good. The source of the oscillation in the original grid system at the fin tip leading and trailing edges appears to be the result of poor inter-grid communication due the minimal grid overlap and large grid resolution discrepancy between the fin and fin tip grids. This problem is further exacerbated by grid overlap occurring close to an explicit axis boundary condition at the fin tip. Such situations have been intentionally avoided in the present grid system. When no sub-iterations were employed, both the original and present grid systems showed stability problems in the boundary layer at the nose singular axis and near the leading edges of the fins.

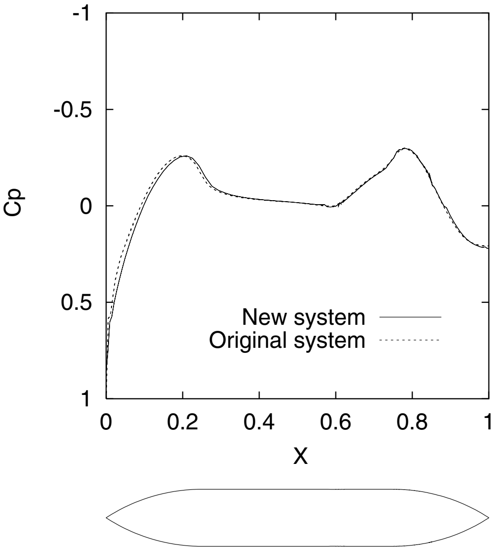

Fig. 13 Surface pressure coefficient along $y = 0$ for the finned-store.

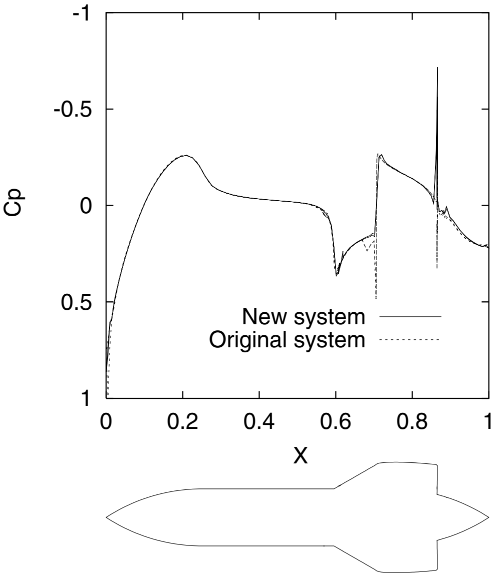

Fig. 14 Surface pressure coefficient along 45 degree sectional cut for the finned-store.

5. Concluding Remarks

A systematic procedure for surface domain decomposition and meshing of complex geometric features using overset structured grids has been presented. The algorithm has been implemented into a software tool called SEAMCR which alleviates the manual effort needed for such a task. The procedure is essentially automatic starting from a multiple panel network description of the geometry. Care must be taken to ensure that the geometry description is trimmed and free of gaps, and that the seam curves meet at coincident points at the corners. Grid spacings on the seam curves and in the spider web grids have been constructed to be less than or equal to a global spacing parameter Δs_g. By adjusting Δs_g, a grid refinement study can be potentially achieved with ease.

Further improvements of the gridding scheme can be accomplished by developing a more sophisticated algorithm for determining marching distances from the seam curves such that more of the geometry surface is covered. More elaborate rules can be added to the seam corner classification procedure so that the singular point grid topology can be avoided for certain classes of situations. Appropriate grid clustering in the radial direction of the spider web grids may be considered based on the surface topology of the seam corner (convex, concave, or saddle).

The new gridding strategy creates surface grids with simple topologies which eases automation of the subsequent steps: volume grid generation, domain connectivity, and flow solution. Results from steady and unsteady flow computations indicate that the new gridding scheme does not appear to cause degradation in accuracy, stability, or convergence, as compared to previous gridding methods. The new procedures show that the entire computational analysis time can be reduced by about half. For more complex configurations, the potential time savings should be even more significant.

Acknowledgements

The authors are grateful to Drs. R. L. Meakin, M. J. Aftosmis, and P. G. Buning for some very helpful and insightful discussions. Geometry for the Hyper-X was provided by the Hyper-X Program Office via Dr. P. G. Buning at NASA Langley Research Center. The original grid system and solution for the X-38 and finned-store were created by J. S. Greathouse of NASA Johnson Space Center, and Dr. R. L. Meakin of the Army Aeroflightdynamics Directorate at NASA Ames Research Center, respectively. The special version of OVERFLOW used on the SGI Origin 2000 at NASA Ames was optimized by J. R. Taft. Funding for this work was provided by the DOD HPCMP CHSSI CFD-4 program and the work was performed under NASA contract number NAS2-14109, Task 23.

References

1. Steger, J. L., Dougherty, F. C. and Benek, J. A., "A Chimera Grid Scheme," *Advances in Grid Generation*, K. N. Ghia and U. Ghia, eds., ASME FED-Vol. 5, June, 1983.

2. Steger, J. L., "Notes on Surface Grid Generation using Hyperbolic Partial Differential Equations," Internal Report TM CFD/UCD 89-101, Department of Mechanical, Aeronautical and Materials Engineering, Univ. of Calif., Davis, 1989.

3. Steger, J. L.,"Grid Generation with Hyperbolic Partial Differential Equations for Application to Complex Configurations," *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, ed. A.S. Ascilla, J. Hauser, P.R. Eiseman and J.F. Thompson, Elsevier Science Publishers B.V. (North-Holland), 1991.

4. Chan, W. M. and Buning, P. G., "Surface Grid Generation Methods for Overset Grids," *Computers and Fluids*, **24**, No. 5, 509–522, 1995.

5. Chan, W. M. and Meakin, R. L., "Advances Towards Automatic Surface Domain Decomposition and Grid Generation for Overset Grids," AIAA Paper 97-1979, in Proceedings of the AIAA 13th Computational Fluid Dynamics Conference, Snowmass, Colorado, 1997.

6. Akdag, V. and Wulf, A., "Integrated Geometry and Grid Generation System for Complex Configurations," *Software Systems for Surface Modeling and Grid Generation*, Ed. Robert E. Smith, Hampton, VA, Langley Research Center, NASA CP 3143, 161–171, 1992.

7. PRO-ENGINEER. See *http://www.ptc.com/products/proe/proe.html.*

8. Steinbrenner, J. P., Chawner, J. R. and Fouts, C. L., "A Structured Approach to Interactive Multiple Block Grid Generation," *AGARD FDP Specialists Mtg. on Mesh Generation for Complex Three-Dimensional Configurations*, Loen, Norway, 1989.

9. Chan, W. M., "OVERGRID - A Unified Overset Grid Generation Graphical Interface," submitted to *Electronic Journal of Grid Generation*, 1999.

10. Chan, W. M., Chiu, I. T., and Buning, P. G., "User's Manual for the HYPGEN Hyperbolic Grid Generator and the HGUI Graphical User Interface," NASA TM 108791, 1993.

11. Suhs, N. E.,and Tramel, R. W., "PEGSUS 4.0 User's Manual," AEDC-TR-91-8, November 1991.

12. Buning, P. G., Jespersen, D. C., Pulliam, T. .H, Chan, W. M., Slotnick, J. P., Krist, S. E. and Renze, K. J., "OVERFLOW User's Manual," Version 1.8, NASA Langley Research Center, 1998.

13. Walatka, P. P., Buning, P. G., Pierce, L., and Elson, P. A., "PLOT3D User's Manual," NASA TM-101067, 1990.

14. Meakin, R. L., "Computations of the Unsteady Flow About a Generic Wing/Pylon/ Finned-Store Configuration," AIAA Paper 92-4568, AIAA Atmospheric Flight Mechanics Conference, Hilton Head Island, South Carolina, 1992.

15. Heim, E. "CFD Wing/Pylon/Finned-Store Mutual Interference Wind Tunnel Experiment," AEDC-TSR-91-P4, January, 1991.

16. Parks, S. J., Buning, P. G., Steger, J. L. and Chan, W. M., "Collar Grids for Intersecting Geometric Components Within The Chimera Overlapped Grid Scheme," AIAA Paper 91-1587 in Proceedings of the AIAA 10th CFD Conference, Honolulu, Hawaii, June, 1991.